

All Empty

Creating package.json = `npm init -y` (all yes)

Done required changes in package.json

Video 2

Done More required changes in package.json

Install express = **`npm i express`** = it will give you package lock.json automatically

Dot env = `npm i dotenv`

Morgan –api req konsa hit hua h show in terminal = `npm i morgan`

Video 3

Create folder config = file db.js for database connection export the function

And use it in server.js

Video 4

Create folder routes,helpers,models,controllers, middlewares

Models/create file userModel.js = skeleton for user database

Now we will create routes

Routes/create file authRoute.js

AuthController/create file authController.js

We use MVC pattern file structure for our project

Model-View-Controller(MVC) architecture for Node applications

`npm i bcrypt`

helpers/create authHelper.js = password hashing

video 5

authController.js = registering user

video 6

npm i jsonwebtoken (see documentation if needed) for login route

video 7

create protected routes for user in middleware using user login token

in database = role 0 = normal user

role 1= admin

tested all the routes with middleware

video 8

working in react

cmd = npx create-react-app client

npm i react-router-dom

VS Code select all

```
<li className="nav-item">
```

Select className do ctrl+ddddd..... for selecting all className variables at once

cmd= npm install react-icons

Video 9

SEO

React by default have no support for SEO it depends on 3rd party apps

Cmd = npm i react-helmet

Removed this by myself not necessary

Video 10 IMP

Pages/create folder Auth/ file Register.js

Have to avoid page refresh on submit =

```
// avoid refresh of page on submit
const handleSubmit = (e) => {
  e.preventDefault();
```

```
console.log(name,email,password,phone,address);  
}
```

npm i axios == handles http network request in frontend and backend

npm i react-hot-toast = notify icons

for frontend protection create .env file in root

In env file = `REACT_APP_API = http://localhost:8080`

Writing "REACT_APP" _youname is the format

npm i concurrently cors == to run two servers concurrently here client and server

Add cors in server js

cmd = PS D:\GitHub Desktop\Ecommerce-App> **npm run**

dev == runs both together

working on register completed

client/src/pages/auth/create Login.js == for login form

video 12

Use **context api** for dynamic Navbar feature using global state = it is also use for large scale apps

Client/src/folder Context/ file auth js

Video 13

Private routes

Client/folder user/file Dashboard js

Client/src/components/ file Private js

Loading Spinner added

userLink location history = after login get back to user left page

video 14

done

video 15

dashboard for user and admin

security

if a normal user tries to access dashboard admin then it will direct the user again to the homepage

components/layout/AdminMenu.js

video 16

for products we will create api in backend

API designing is done here for products

npm i slugify = for whitespace type of link eg user/**category-shirts**

These types of APIs are commonly referred to as ****RESTful APIs**** (Representational State Transfer APIs).

RESTful APIs adhere to the principles of REST, which is an architectural style for designing networked applications. RESTful APIs use standard HTTP methods (such as GET, POST, PUT, DELETE) to perform CRUD (Create, Read, Update, Delete) operations on resources.

RESTful APIs are widely used due to their simplicity, scalability, and flexibility, making them a popular choice for building web services and integrating different systems.

Video 17 **(Hard video see more)**

Product API

npm i express-formidable = for handling photo checker

all the CRUD on database for product operation of the backend implemented here

video 18 **(Hard video see more)**

form folder created in components

Ant design web for components of form 16:00

client cmd = npm install antd

video 19 **(Hard video see more)**

uploading photo option for a product 10:00

video 20 Just see

video 21

6:21

Here we designed the filter and category option to work on home page important

Main work 22:00

We cant load all the products at once it will overload our api so we will apply pagination to avoid failure

Video 22

Will try to implement search box globally

Create a global keyboard state to get it = context API will be used

React parameters are used in React routing, where we have parameters we need to access in the route. For example, if we had a route such as `<Route path="/:id" />` we could access that particular string or value in the route by calling the `useParams` hook. `let { id } = useParams();`

The `$or` operator in MongoDB is a logical operator that performs a logical OR operation on an array of two or more expressions and selects the documents that satisfy at least one of the expressions.

The Context API in React is a feature that allows you to share data between components without having to explicitly pass props through every level of the component tree. It provides a way to pass data through the component tree without

having to pass props down manually at every level. This can be particularly useful for passing global data or state down

to deeply nested components.

Here's a basic overview of how the Context API works:

Create a Context: You first create a context using

the `React.createContext()` function. This creates a Context object that contains a Provider and a Consumer component.

Provide Data: You wrap the part of your component tree where you want to provide data with a Provider component. This Provider component accepts a value prop,

which can be any JavaScript value.

Now, regarding the difference between Context API and REST API:

Context API is a

feature provided by React for managing state within a React application. It's used for managing application state within the client-side of

the application.

REST API stands for Representational State Transfer and is an architectural style for designing networked applications. RESTful APIs are typically

used for communication between client-side applications (such as a React application) and server-side applications (such as a database or web server).

REST APIs allow you to perform operations like retrieving, updating, creating, and deleting resources (such as data records) over HTTP.

In summary,

Context API is used for managing state within a React application, while REST API is used for communication between client-side and

server-side applications over HTTP. They serve different purposes and are used in different parts of the application stack.

Implemented similar products section

Video 26

We are using **Braintree** for payment gateway

Make account in developer Paypal also or use dummy

Server cmd = npm i Braintree

Client/react cmd = npm i braintree-web-drop-in-react

Video 27

Client/cmd = npm i moment for accessing current time and date

